

APPENDIX AHeuristics Stylesheet:

```
<?xml version="1.0" encoding="utf-8" ?>
```

5

```
<xsl:stylesheet
  version="1.0"
  extension-element-prefixes="saxon"
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'
  xmlns:outxsl='output.xsl'
```

10

```
  xmlns:out='default.xsl'
>
```

```
<xsl:output indent="yes"/>
```

15

```
<!-- Declare namespace aliases. When we write out the created
  -- stylesheet, "out" will map to the default (empty) namespace and
  -- "outxsl" will map to the "xsl" namespace.
-->
```

```
<xsl:namespace-alias stylesheet-prefix="out" result-prefix=""/>
```

20

```
<xsl:namespace-alias stylesheet-prefix="outxsl" result-prefix="xsl"/>
```

```
<!--
```

25

At the beginning of the generated stylesheet, we need to emit an <xsl:stylesheet> and, for convenience, we declare a global variable to hold the representation of a newline symbol. We also set the <xsl:output> to indent="yes" in the generated stylesheet to get a prettier output. Then, in the generated stylesheet, emit a template to match on the <document> element and within that we apply-templates to the <line> children.

30

```
-->
```

```
<xsl:template match="/">
```

```
  <outxsl:stylesheet version="1.0">
```

```
<outxsl:variable name="nl" select="
"/>
<outxsl:output indent="yes"/>
```

5

```
<!-- Emit a template for <document>. Within that template,
-- apply-templates to the <line> children then emit a newline
-- for prettyprinting purposes.
-->
```

10

```
<outxsl:template match="document">
  <out:nitf>
  <outxsl:apply-templates select="line"/>
  <outxsl:value-of select="$nl"/>
</out:nitf>
</outxsl:template>
```

15

```
<!-- Emit a template for <line>. The xsl:apply-templates will
-- cause the various <xsl:when> elements to be emitted. After
-- the <xsl:when> clauses have been emitted, we emit an
-- <xsl:otherwise> clause that simply emits a newline character.
-->
```

20

```
<outxsl:template match="line">
  <outxsl:choose>
    <xsl:apply-templates/>
    <outxsl:otherwise>
      <outxsl:value-of select="$nl"/>
    <outxsl:apply-templates/>
  </outxsl:otherwise>
</outxsl:choose>
</outxsl:template>
```

25

30

```
</outxsl:stylesheet>
</xsl:template>
```

```

<!-- process the headline heuristic -->
<xsl:template match="headline">
  <outxsl:when>
    <xsl:attribute name="test">
5      <xsl:value-of select="concat('position()='',./line)"/>
    </xsl:attribute>
    <out:hedline>
      <outxsl:apply-templates/>
    </out:hedline>
10    </outxsl:when>
  </xsl:template>

<!-- process the byline heuristic -->
<xsl:template match="byline">
15  <outxsl:when>
    <xsl:attribute name="test">
      <xsl:value-of select="concat('position()='',./line)"/>
    </xsl:attribute>
    <out:byline>
20    <outxsl:apply-templates/>
    </out:byline>
    </outxsl:when>
  </xsl:template>

25  <!-- process the dateline heuristic -->
  <xsl:template match="dateline">
    <xsl:choose>
      <xsl:when test="./line">
        <outxsl:when>
30          <xsl:attribute name="test">
            <xsl:value-of select="concat('position()='',./line)"/>
          </xsl:attribute>

```

```

        <out:dateline>
        <outxsl:apply-templates/>
    </out:dateline>
</outxsl:when>
5 </xsl:when>
  <xsl:when test="./string">
    <xsl:choose>
      <xsl:when test="./string[@pos='first']">
        <outxsl:when>
10      <xsl:attribute name="test">
        <xsl:value-of
          select="concat('starts-with(.,&quot;',./string,&quot;)'")"/>
        </xsl:attribute>
        <out:dateline>
15      <outxsl:apply-templates/>
        </out:dateline>
        </outxsl:when>
      </xsl:when>
    </xsl:choose>
20 </xsl:when>
  </xsl:choose>
</xsl:template>

<!-- process the paragraph heuristic -->
25 <xsl:template match="paragraph">
  <xsl:choose>
    <xsl:when test="blank/@type='line'">
      <outxsl:when>
        <xsl:attribute name="test">
30      <xsl:text>.="</xsl:text>
      </xsl:attribute>
      <outxsl:value-of select="$nl"/>

```

<out:p/>

&lt;/outxsl:when&gt;

&lt;/xsl:when&gt;

&lt;/xsl:choose&gt;

&lt;/xsl:template&gt;

&lt;/xsl:stylesheet&gt;

APPENDIX BDiscrimination Stylesheet:

```

5  <?xml version="1.0" encoding="utf-8" ?>
   <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
     version="1.0">
     <xsl:variable name="nl" select="' ' ' " />
     <xsl:output indent="yes" />
10  <xsl:template match="document">
     <nitf>
       <xsl:apply-templates select="line" />
       <xsl:value-of select="$nl" />
     </nitf>
15 </xsl:template>
    <xsl:template match="line">
      <xsl:choose>
        <xsl:when test="position()=1">
          <hedline>
20       <xsl:apply-templates />
          </hedline>
        </xsl:when>
        <xsl:when test="position()=2">
          <byline>
25       <xsl:apply-templates />
          </byline>
        </xsl:when>
        <xsl:when test="starts-with(., '[DATE]')">
          <dateline>
30       <xsl:apply-templates />
          </dateline>
        </xsl:when>

```

```
<xsl:when test=".=<"">
    <xsl:value-of select="$nl" />
    <p />
</xsl:when>
<xsl:otherwise>
    <xsl:value-of select="$nl" />
    <xsl:apply-templates />
</xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:stylesheet>
```

